

Ajuste na listagem dos alunos

Transcrição

[00:00] Agora que nós já estamos conseguindo recuperar os alunos que nós estamos persistindo no nosso device, através do Core Data, havia um problema que nós estávamos discutindo na última aula, que era o seguinte, quando nós vínhamos no formulário e cadastrávamos um novo aluno, quando nós clicamos em “salvar”, ele nos redireciona pra home, porém o aluno não estava aparecendo pra nós automaticamente, que é o que deveria acontecer.

[00:24] Eu cadastrei um novo aluno, ele já existe no nosso banco, ele está no nosso device, porém ele não mostra aqui na table view, ele só aparece se nós reiniciamos o app, o rodamos de novo e tudo mais. Agora é hora de nós resolvermos esse problema que nós temos.

[00:41] Quando nós utilizamos o Core Data, nós vimos que para trazer os resultados nós utilizamos a classe “FetchedResultsController”. Essa classe, tem alguns métodos de delegate também que nós podemos utilizar, e um deles é um que fica observando se houve alguma mudança ou, por exemplo, se eu delete um aluno é uma mudança, se eu edito um aluno, se eu cadastro um novo aluno, tudo isso ele observa pra nós.

[01:09] Nós podemos utilizar esse método pra ficar observando e nós tomamos alguma decisão conforme o que precisamos. No caso, quando eu cadastrar um novo aluno, eu preciso que ele apareça na table view pra nós visualizarmos, então é através desse método que nós vamos conseguir resolver esse problema.

[01:26] O primeiro passo então, quando nós vamos trabalhar com algum método de delegate, é utilizar o seu protocolo, então eu vou vir aqui e vou colocar “NSFetchedResultsControllerDelegate”, é essa opção aqui que nós vamos pegar.

[01:44] Quando nós implementamos esse protocolo, nós já temos acesso ao método de delegate dessa classe. Porém, para funcionar, nós precisamos setar o delegate no nosso View Controller, ou seja, eu tenho que falar que quem vai implementar os métodos de delegate do Results Controller é esse View Controller e nós fazemos isso setando o seu delegate.

[02:05] Então eu vou vir aqui, colocar “gerenciadorDeResultados.delegate = self”, ou seja, é o “HomeTableViewController” que vai implementar o protocolo de delegate e os seus métodos que nós vamos utilizar.

[02:22] Feita essa configuração, nós já conseguimos utilizar esse método, que é o que nós vamos fazer agora. Porém, repare que nós estamos separando todo o código com esses marks, eles servem pra nós identificarmos mais rápido. Então se eu clicar nesse cara aqui, por exemplo, ele exibe todos os marks que tem no nosso código. Eu quero ver os métodos, eu venho aqui, ele traz pra métodos. Eu quero ver o view lifecycle, eu clico aqui e vejo todos os métodos.

[02:50] Pra ficar mais organizado, nós vamos fazer a mesma coisa aqui, nós vamos criar um novo marcador pra esse método de delegate do Fetch Request. Então vou pôr aqui “FetchedResultsControllerDelegate”, vou deixar tudo junto, e nós vamos implementar o método que nós vamos utilizar para ficar observando alguma mudança no Core Data.

[03:15] Qual é o método? Eu só vou dar um espaço aqui pra ficar com a visualização melhor. Vou começar a digitar o fetched, ele vai nos trazer alguns métodos. Repare que parece que é tudo igual tem controller, controller, controller aqui, só que no final tem algumas diferenças. O que nós vamos precisar, é esse método aqui, “NewIndexPath”, onde ele vê se ocorreu alguma mudança.

[03:45] Vou clicar, “didChange”, então ele vai ficar observando alguma mudança. Quando ocorre alguma mudança, nós podemos tratar, dependendo do evento que aconteceu, então eu posso pôr aqui um “switch”, aí o “type”.

[04:01] Pra ficar mais fácil, eu só vou fazer uma ação específica quando for o deletar, então eu posso tratar aqui, se for delete eu vou implementar algum código que nós já vamos ver, vou dar um “break”. E se for qualquer outro tipo, sem ser o delete, eu vou dar um reload na nossa table view. Então vou pegar o “tableView.reloadData”, é isso que eu preciso fazer nesse momento, dar o reload. E quando for delete, nós vamos implementar daqui a pouco.

[04:31] Feito isso, nós vamos rodar o aplicativo e vamos testar essa implementação, então vamos lá. Vou dar um rodar aqui no nosso app. Então vamos adicionar um novo aluno? Eu vou clicar aqui no “+”, vir em foto, vamos escolher qualquer foto mesmo, e eu vou cadastrar agora o Lucas. “Lucas”, ele mora em Bragança Paulista, aqui tem o telefone dele, o site vai ser “www.alura.com.br” (<http://www.alura.com.br%E2%80%9D>) e eu vou dar uma nota para esse aluno, que é o 8. Vou clicar aqui em salvar.

[05:08] Ele está trazendo já pra nós, instantaneamente, o Lucas. O que está acontecendo? Quando nós cadastramos um novo aluno, ele cai nesse método, que é o “didChange” e dá um reload na nossa table view, por isso que está aparecendo agora o aluno, logo quando nós cadastramos.

[05:24] Agora, pra fechar essa parte de listagem, repare que a imagem aqui está quadrada. Vamos arredondá-la pra ficar mais legal o nosso layout? Eu vou criar aqui um método na classe onde fica a célula, pra nós tirarmos do View Controller eu vou deixar toda a personalização da célula aqui. Então vou criar um método, “func configuraCélula()”, e eu vou passar o aluno como parâmetro, então “aluno” é do tipo “Aluno”.

[05:59] Só lembrando, eu coloco esse “_” aqui pra na hora que eu for utilizar o método, não ficar duplicado o nome, tipo aluno e aluno, então aqui nós ocultamos o parâmetro interno do método.

[06:12] Agora eu vou vir no “HomeTableViewController” e vou utilizar aquele método, vou chamar “celula.configuraCélula”, então aqui eu posso passar já direto o aluno, sem duplicar o nome no parâmetro.

[06:30] Agora toda essa parte de configuração da célula eu vou deixar pra aquela célula ficar encarregada de fazer isso. Então vou vir aqui, vou colar esse método. Como eu estou dentro da própria classe da célula, eu não preciso da referência dela, aqui também eu não preciso da referência da célula.

[06:51] Agora, pra arredondar, o que nós vamos fazer? Eu vou chamar a “imageAluno.layer.cornerRadius” é igual a própria imagem “.frame.”, a largura, por exemplo, e vou dividir por 2, eu consigo arredondar.

[07:07] Feito isso, eu vou fazer o seguinte, eu vou pegar um layer de novo, vou pegar esse “maskToBounds” e vou passar o “true”, porque é a marca de recorte que nós precisamos fazer. Agora nós vamos rodar o app de novo e vamos testar.

[07:23] Nós já estamos conseguindo fazer aparecer automaticamente o novo aluno quando nós o cadastramos no formulário, agora falta aprender como faz pra nós editarmos algum dado desse aluno e também apagar, então faltam essas duas operações pra nós vermos como fazemos com o Core Data. Já nós vamos ver isso.